BRL

MEMORANDUM REPORT NO. 1379
NOVEMBER 1961

THE INSTRUCTION CODE FOR THE BRL ELECTRONIC SCIENTIFIC COMPUTER (BRLESC)

Lloyd W. Campbell
Glenn Beck

Department of the Army Project No. 503-06-002
Ordnance Management Structure Code No. 5010.11.812
BALLISTIC RESEARCH LABORATORIES

ABERDEEN PROVING GROUND, MARYLAND

BALLISTIC RESEARCH LABORATORIES

MEMORANDUM REPORT NO. 1379

NOVEMBER 1961

THE INSTRUCTION CODE FOR THE BRL ELECTRONIC SCIENTIFIC COMPUTER (BRLESC)

Lloyd W. Campbell
Glenn Beck

Computing Laboratory

Department of the Army Project No. 503-06-002
Ordnance Management Structure Code No. 5010.11.812

ABERDEEN PROVING GROUND, MARYLAND

BALLISTIC RESEARCH LABORATORIES

MEMORANDUM REPORT NO. 1379

LWCampbell/GABeck/bjw
Aberdeen Proving Ground, Md.
November 1961

THE INSTRUCTION CODE FOR THE BRL ELECTRONIC SCIENTIFIC COMPUTER (BRLESC)

ABSTRACT

The complete instruction repertoire of the BRL's new Electronic Scientific Computer (BRLESC) is given. Some engineering features are included.

TABLE OF CONTENTS

I. INTRODUCTION

BRLESC is a large, high speed, electronic computer that is now in operation at the Ballistic Research Laboratories. It has a relatively simple but very effective instruction repertoire that has been designed for ease of programming, speed, and a high performance-to-cost ratio. It also has many design features that help to simplify the task of writing programs for this machine. Some of these desirable design features are:

1. Three-address instructions.
2. Sixty-three index registers.
3. Index registers that are individually addressable as part of the main memory, and have simultaneous access with the main memory.
4. Index registers that modify addresses by addition, not subtraction.
5. A good set of logical instructions that perform a variety of boolean operations.
6. Simple and fast instructions for doing "book keeping" operations concurrently with arithmetic and logical instructions.
7. An instruction for easy use of subroutines.
8. Input-output that is concurrent with computing and with other input-output devices.
9. Automatic built-in checks and interlocks on almost all of the concurrent operations.
10. Fast floating point arithmetic, almost as fast as fixed point arithmetic.

Because of these design features and the instruction repertoire, programmers find it relatively easy to communicate with BRLESC and it is relatively easy to translate from a formula type language into BRLESC instructions.

II. ARITHMETIC AND LOGICAL UNIT

	Microseconds	
Operation	Excluding Access Time	Including Access Time
Fixed point add or substract	1	5
Fixed or floating multiply	20	25
Fixed or floating divide	60	65
Floating add or subtract	3	6
Boolean logic operation	1	5
Indexing and control	2	6

Arithmetic mode	Parallel
Timing	Synchronous

Construction

The arithmetic unit is constructed of standard vacuum tube logical packages, with tube driven, crystal diode logical gating. The arithmetic unit is constructed of 1,727 vacuum tubes of 4 types, 853 transistors of 3 types, 46,500 diodes of 2 types and 1,600 pulse transformers of 1 type.

Logical events are controlled by a five-phase clock, permitting decisions at the rate of five million per second.

III. NUMERICAL SYSTEM

Internal number system	Binary
Binary digits per word	68 + 4 parity
Binary digits per instruction	68
Instructions per word	1
Instructions decoded	33
Arithmetic system	Fixed and floating point
Instruction type	Three-address

IV. WORD FORMAT

Instruction Format

4	4	6	14	6	14	6	14
Order type	Parameter	Index a	α-Address	Index b	β-Address	Index c	γ-Address

Fixed Point Number Word Format

3	1	4	•	60
Tag	Sign	Bits	Binary Point	Bits

Floating Point Number Word Format

3	1	4	•	52	8
Tag	Sign	Bits	Binary Point	Bits	Biased Exponent of 16
		Coefficient			

V. FORMULA TRANSLATION

In order to equip relatively inexperienced programmers as well as experienced programmers with an easy, simple, and rapid method for preparing their problems for BRLESC solution, a formula translator has been devised, called FORAST. When FORAST is used it is not necessary to understand or use the list of programming instructions given below, and a knowledge of the binary system of notation is not required. A description of the method will be published in a BRL memorandum report.

VI. PROGRAMMING SUMMARY

BRLESC has 63 one-microsecond cycle-time index registers, addressable by the a, b, and c addresses of the instruction words.

The parameter bits of the instruction word are used to indicate variations of the basic order type.

All three arithmetic registers are 68 bits long. Tag bits enter these registers only on the logical instructions and the shift instructions, provided it is a cyclic or a logical shift. On arithmetic orders, the tag bits are saved in a separate three-bit register and the three extra bits in the arithmetic registers are used for checking overflow.

The three tag bits may be manipulated logically by the programmer. The tag bits of an arithmetic result will have a "one" in any position that was a "one" in any of the operands involved. All 68 bits are used in the instructions.

All numbers use two's complement representations. Thus, fixed point numbers and floating point coefficients have the range $-16 \leq N < 16$. A floating point number has a biased exponent. (Bias = 128) Floating point numbers are of the form: Coefficient x $16^{\text{Exponent + Bias}}$.

When an instruction is performed, an effective address is computed by adding the contents of the specified index register to the specified memory address. We shall refer to the three effective addresses as A, B, and C, i. e., $C = (c) + \gamma$ where (c) denotes the contents of memory or index register c, and γ is the address contained in the instruction being executed. An index of 0 means no index (i. e. if $a = 0$; $A = \alpha$).

The 63 index registers are provided in a separate core memory and they are also addressable in α, β, or γ by addresses 1 to 63. They are 14 bits long (expandable to 16 bits if needed later) and are used as the 14 least significant bits when addressed as full words. There is no 0 index memory position.

Fourteen bit addresses allow direct addressing up to 16,384 words, however the present memory capacity is 4,096 words.

Memory position 0000 is a special register that gets the result of each arithmetic or logical instruction. Succeeding instructions may use the contents of the special register whenever a α or b β is 00000. A store address of 00000 will cause the result to be placed only in the special register.

VII. INSTRUCTION LIST

The parameter bits P are used on arithmetic instructions to indicate possible variations of the basic order type. Call these 4 bits, P_{64}, P_{63}, P_{62}, P_{61}, then the following variations are indicated by these bits:

		Symbolic Par.	
P_{61}	0	F	Do floating point arithmetic
	1	X	Do fixed point arithmetic.
P_{62}	0		Do not accumulate in C
	1	A or +	Accumulate in C (i. e. Add result to (C) before storing in C).
P_{63}	0		Use algebraic value of operands
	1	V	Use absolute value of operands. (Could be $/R/ + /A/ - /B/ + /C/ \rightarrow (C)$).

P_{64}	0	N	If floating point, assume normalized operands and produce normalized result.
	0	N	If fixed point, ignore previous contents of R register.
	1	U	If floating point, do unnormalized significant digit arithmetic.
	1	R	If fixed point, use contents of R register. (Adds (R) to (A) and then does operation except on multiply and divide.)

Note: In FORAST N implies normalized fl[illegible] [illegible]

Arithmetic Instructions

Sexa. O. T.	Symbolic O. T.	Name	Brief Description
2	A or +	Add	(A) + (B) → (C)
3	S or -	Substract	(A) - (B) → (C)
4	M or *	Multiply	(A) x (B) → (C)
5	D or /	Divide	(A) ÷ (B) → (C)
6	C or C-	Compare	If (A) < (B) and P_{62} = 0
	or C+		or if (A) ≥ (B) and P_{62} = 1
			next instruction = (C)
7	SQ or SQ RT	Square Root	$\sqrt{/(A)/} \rightarrow$ (C)
8	SH	Shift	(A) shifted by shift code B → (C) (See Shift Code.)

[illegible] P_{64}

Logical Instructions

9	TP	Transplant [illegible] Let S=11, [illegible] a_{51} [illegible] [illegible] 51 ([illegible] [illegible] (B) is called a "mask". [illegible] a left 44 [illegible] [illegible] 64 [illegible] [illegible] cyclic [illegible] a_{51} [illegible] [illegible] c_{51} of the mask is 1 or 0, c_{31} [illegible] equal to a_{51} [illegible]	$[(A)$ AND $(B)]_s$ + $[(\overline{B})s$ AND $(C)] \rightarrow (C)$ where subscript s indicates that the word has been cyclic shifted left 4P (P = Parameter) bits and $(\overline{B})$ is digitwise complement of (B). "AND" is a logical bit-by-bit product. The tag and sign bits of all words are involved in the logical operations but are not included in the cyclic shift path. Note that this instruction allows bits in (C) to be replaced with bits taken from (A) that are in positions where (B) contains a 1.
K	B	Boolean	Generate the boolean function of (A) and (B) as specified by P and store it in C. (See Boolean Function Table)
S	CB	Compare Boolean	If the boolean function of (A) and (B) as specified by P is zero, next instruction = (C).
N	CNB or CN	Compare Boolean for Not Zero	If the boolean function of (A) and (B) as specified by P is not zero, next instruction = (C).
J	PMA	Polynomial Multiply	(A) x (B) + (C) → (000) (must use accumulate bit P_{62} = 1.)

Special Interpret Instruction

L	IT	Interpret	~~Save the address of this instruction in index register 1~~ and take next instruction = (64) dec. (040) sexa.*

Indexing and Control Instructions

Basic Order Type = 0 or 1; P is used as part of Order Type.

Sexa. O. T.	Symbolic O. T.	Name	
00	HALT	Halt	Halt machine (except for input-output trunks), then if initiated, next instruction = (C).
01	SET or SI	Set Index	$\alpha \rightarrow (a)$; $\beta \rightarrow (b)$; $\gamma \rightarrow (c)$. Sets up to three index registers from α, β, γ.
02	INC or II	Increase Index	$A \rightarrow (a)$; $B \rightarrow (b)$; $C \rightarrow (c)$. Increase up to three Index registers by increments written in α, β, γ. Negative increments may be used by using 2's complement increments. (As long as the memory address does not exceed 14 bits).
03	LP or LOOP	Loop	$\alpha + 1 \rightarrow (a)$ if $a \neq 0$. Then if $\alpha + 1 < B$, restore instruction with $\alpha + 1 \rightarrow \alpha$ and take next instruction = (C). But if $\alpha + 1 \geq B$, restore instruction with $\alpha = 0$ and go on to next sequential instruction.

* [illegible] "Jump [illegible] address [illegible] I.K. 7"

04	J or JUMP	Jump	Next instruction = (C).
05	JS	Jump to Subroutine	Save address of this instruction in index 1 ~~and~~ $A \rightarrow (a)$; $B \rightarrow$ [illegible] and do jump to (C). (W[illegible] [illegible])
06	J+	Jump if +	Jump to (C) if (A) $\geq$ 0. (See Concurrent Operation.)
07	J-	Jump if -	Jump to (C) if (A) $<$ 0. (See Concurrent Operation.)
08	CARD TAPE ~~DRUM~~ (Large memory [illegible]) FLEX IO	Input-Output	Transfer B words into or out of consecutive memory positions starting at A. $c\gamma$ specifies type of transfer etc. (See page 22)
09	SIJ or SETJ	Set Index and Jump	$\alpha \rightarrow (a)$; $\beta \rightarrow (b)$ and jump to (C).
OK	IIJ or INC J	Increase Index and Jump	$A \rightarrow (a)$; $B \rightarrow (b)$ and jump to (C).
OS	EA	Compute Effective Addresses	$A \rightarrow b$ and $C \rightarrow \beta$ where β must be index memory address.
ON	JA	Jump if Alphabetic Character	Jump to (C) if rightmost 6 bits of (A) are same as rightmost 6 bits of β.
OJ	JC	Jump if Condition	β contains a code which determines which overflow or other conditions are to be tested. If any tested condition is on, it is turned off and jump to (C) is done.

OF	NOP	No operation	Proceed to next instruction.
OL	RSW	Manual Read	Read contents of 68 switches on console and store that word in B memory positions beginning at A. Then jump to (C).
13	LPI or LOOP I	Loop on Index	Same as 03 Loop order except A (instead of α) is compared with B and the instruction is never restored. (The count is kept only in index register a.)
IN	JNA	Jump if Not Alphabetic character	Jump to (C) if rightmost 6 bits of (A) are not equal to rightmost 6 bits of β.
1J	JNC	Jump if Not Condition	Same as Jump if condition (OJ) except it jumps when none of the conditions tested are "on".

There are three more jump instructions which are being considered for future use with "permanent" storage of "built-in" subroutines. It is planned that subroutines will be stored in a non-erasable bank of memory whose addressing scheme is such that reference to most of the main memory as well as reference to itself within 256 word sections can be made by such "built-in" subroutines.

14	JB	Jump to "Built-in" Instruction	Jump to (C) where C is address within "permanent" memory.
15	JSB	Jump to "Built-in" Subroutine	Same as 05 instruction except C is address within "permanent" memory.
19		Set Index and Jump to main memory	Same as 09 instruction except next instruction is always from main memory.

10 MMF Move B words from A, A+1, etc. to C, C+1, etc.
B may be zero. Neither A nor C may be an index register.
18 MMB Same except "from A, A-1, etc. to C, C-1, etc."

Write-up of 9 Oct 68 "BRLESC II FORAST". p 25.

Additional Comments on Parameter Bits and Arithmetic Instructions

A. Rounding

All floating point instructions give rounded results except compare. Shift will round or not, depending on P_{31} in the shift code.

Fixed point multiply and divide are rounded only if $P_{64} = 0$.

If $P_{64} = 1$ on fixed point multiply, as exact double length product is obtained and the least significant part is left in the R register. The bits to the right of the point in R are the least significant part of the product. The four bits in front of the point are a duplication of the last 4 bits of (C) and the sign bits (there are 4 "sign" bits in all registers of the arithmetic unit) are all set to the sign bit of the most significant part.

If $P_{64} = 1$ on Fixed Point divide, the bits to the right of the point in R (left there by previous order) are used as the least significant bits of the dividend. (The sign of R is ignored). Also the quotient is not rounded and the remainder is left in the R register.

B. Overflow and Underflow

For normalized Floating Point divide, the divisor must be a normalized non-zero number. If it isn't, the divide by zero overflow is turned on and the dividend coefficient x 16^{127} is stored as the result. A very large number is stored as the result of any Floating Point order that causes exponent overflow. (Exp. = 127.) Zero is stored anytime exponent underflow occurs. (Exp. = -128.)

Except for Fixed Point divide, numbers that result when Fixed Point overflow occurs will have the proper sign and answer (mod 16) if the result is ≥ -128 and < 128. A left shift will always give proper sign and answer (mod 16).

Whenever any type of overflow occurs, a flip-flop is turned "on" and is turned off only when tested by "Jump if condition" or "Jump if not condition" orders. There will be optional manual switches that will cause the computer to stop at the instruction that caused the overflow.

C. Tag Bits

All three arithmetic registers are 68 bits. Tag bits enter these registers only on the logical instructions and the shift instruction if it is cyclic or is a logical shift ($P_{33} = 1$ in shift code). On arithmetic orders, the tag bits are saved in a separate three bit register and the three extra bits in the arithmetic registers are used for checking overflow. Thus the range of numbers in the arithmetic unit is $-128 \leq N < 128$.

D. Unnormalized Significant Digit Floating Point Arithmetic

Add and subtract are performed same as for normalized arithmetic except the result is never shifted left at the end of the operation.

Before multiply is done, the coefficient that has the largest absolute value is normalized. The result is normalized left a maximum of one sexadecimal place. Thus the result has approximately the same number of significant digits as the operand that had the smaller number of significant digits. It does tend to retain an average of about two more bits than it should however.

Before divide is done, both operands are normalized but the number of divide steps performed is reduced accordingly so that the result has approximately the same number of significant digits as the operand that had the smaller number of significant digits.

Before square root is done, the operand is normalized and the number of steps is reduced accordingly so that the result has about the same significance as the operand.

On floating point shift instructions, the unnormalized bit ($P_{64} = 1$) has a somewhat different meaning. It means do a floating point type shift of the amount specified in the shift code, even if bits get shifted off scale.

E. <u>Shift Code</u>

B digits		Symbolic	
([illegible] of P61) P_{21-27}			Magnitude N of shift where $0 \leq N < 127$. N = no. of sexadecimal digits to shift on floating point. N = no. of bits to shift on fixed point. ($P_{61} = 0$ fixed point, $P_{61} = 1$ floating point)
P_{28}	0	L or +	Left Shift
	1	R or -	Right shift
P_{29}	0		normal
	1	Z	Clear rightmost 8 bits of word to 0 before it is shifted.
P_{30}	0		Non-Cyclic
	1	C	Cyclic (Assume $P_{31} = P_{34} = 0$ and $P_{33} = 1$) [illegible]
P_{31}	0		No rounding
	1	R	Round (Add 2^{-60}, or 2^{-52} for floating point, if last bit shifted off scale to right was 1)
P_{32}	0		Do not include sign and tag bits in shift path.
	1	T	Include sign and tags in shift path.
P_{33}	0		Number Shift; Spread sign on right shift - check overflow on left shift - Save and reset tag bits as on arithmetic instructions (Assume $P_{32} = 0$) (See $P_{30} = 0$ comment on next page)
	1	B	Boolean (logical) Shift; Put tag bits in register - insert 0's when shift is non-cyclic - no check for overflow

P_{34}	0		normal
	1	D	Double length shift; Shift both (A) and (R). On fixed point, shift between A_{1-4} and R_{57-60}. On floating point shift between A_{9-12} and R_{57-60}.

Additional comments on floating point shift order:

Floating point shift assumes $P_{30} = P_{32} = P_{33} = 0$. (Always shifts non-cyclic number.) Parameter bit P_{64} has special meaning on floating point shift. If $P_{64} = 0$ on a left shift, the number is never shifted off scale. Shifting stops if the number becomes normalized. If $P_{64} = 1$, the shift is done exactly as specified and digits of the coefficient may be shifted off of either end. Only the coefficient is shifted on a floating point shift and the exponent is adjusted so that the value of the number will not change. (Except when the shift is one that causes digits to be lost.) An absolute value left shift of magnitude 0 and with $P_{64} = 0$ will cause a right one shift if the original coefficient was -16.

Boolean Function Table for Boolean, Compare Boolean, and Compare Boolean for not Zero Instructions

Let (A) = x and (B) = y and $\bar{x}$ be digitwise complement of x. All words are 68 bits.

Let xy signify the logical "and" operation on x and y, (bit by bit product) and x + y signify the logical inclusive "or" operation on x and y.

Parameter Value	Function	Parameter Value	Function
0	0	8	$x\ y$
1	$\bar{x}\ \bar{y}$	9	$x\ y + \bar{x}\ \bar{y}$
2	$\bar{x}\ y$	K	y
3	$\bar{x}$	S	$\bar{x} + y$

4	$x \bar{y}$	N	x
5	$\bar{y}$	J	$x + \bar{y}$
6	$x \bar{y} + \bar{x} y$	F	$x + y$
7	$\bar{x} + \bar{y}$	L	1

Function 6 gives 0 if and only if (A) = (B) and is "exclusive or".
Function 8 is logical "and" and F is inclusive "or".
Function 0 is fastest way to clear a memory position.

Input-Output Equipment

1. A 1402 IBM Card Reader-Punch Unit that can read 800 cards per minute and punch 250 cards per minute.

2. Six magnetic tape handlers that have a transfer rate of approximately 120,000 six-bit characters per second. A one-inch tape that has sixteen channels is used so that it is possible to have two six-bit characters per row. The block length is variable. There are two tape "trunks" that can simultaneously transfer information between the memory and two tape handlers.

3. A flexowriter that can be used for printing short error and operator messages. It prints ten characters a second.

4. Peripheral Equipment. There is a single off-line unit that is capable of transferring alphanumerical six bit characters from (a) cards to tape (b) tape to high speed printer (1200 lines per minute) (c) tape to cards (d) cards to high speed printer (e) tape to tape and (f) paper tape (8 channel) to magnetic tape.

~~5. Future plans include the addition of five magnetic drums as soon as possible. Each drum will have a total of three channels and a capacity of at least 6,144 words. The transfer rate will be about 8 microseconds per word with an average access time of about 12 milliseconds for the first word.~~ Large memory [illegible]

cγ for Input-Output Orders

Let cγ (not indexable) = $c_5 \; c_4 \; c_3 \; c_2 \; c_1$ where each c_i is one sexadecimal digit.

C_1 sexadecimal	Symbolic	
0	PUNCH 68	Card punch 68 columns. (One word per row).
1	READ 68	Card read 68 columns. (One word per row).
2	W - uA or W_{10} - uA	Write on magnetic tape 60 bits per word using trunk A. (Uses rightmost 60 bits of word.) (u is tape unit number)
3	R - uA or R_{10} - uA	Read from magnetic tape 60 bits per word using trunk A.
4	W - uB or W_{10} - uB	Write on magnetic tape 60 bits per word using trunk B.
5	R - uB or R_{10} - uB	Read from magnetic tape 60 bits per word using trunk B.
6	DW	Write on drum.
7	DR	Read from drum.
8	PUNCH 80	Card punch 80 colums. (Two words per row, 68 + rightmost 12 bits of second word.)
9	READ 80	Card read 80 colums. (Two words per row).
K	W_{12} - uA	Write on magnetic tape 72 bits per word (68 + 4 parity bits) using trunk A.
S	R_{12} - uA	Read from magnetic tape 72 bits per word using trunk A.
N	W_{12} - uB	Write on magnetic tape 72 bits per word using trunk B.
J	R_{12} - uB	Read from magnetic tape 72 bits per word using trunk B.

C_2 sexadecimal

Number of tape unit (1-15) for tape orders.
(Unit 0 is the flexowriter).
Drum channel for drum orders.

C_3 sexadecimal

(Apply to tape orders only.)

Tape reading:

0	R or R_{10} or R_{12}	Read a maximum of one block even if it contains fewer than B words.
1	RN or RN_{10} or RN_{12}	Read as many blocks as are required to transfer B words to memory. NOTE: When reading, tape always moves to the end of the block if the count B reaches 0 in the middle of a block.
2	MF	Move tape forward B blocks.
3	MB	Move tape backward B blocks.
4	UNW	Unwind tape forward. (Move to end of tape.)
5	REW	Rewind tape backward. (Move to beginning of tape.)
6	REWI	Rewind and interlock tape unit.
7	MFMB	Move to file mark backward.
8	MFMF	Move to file mark forward
K	PR	Prepare to read beginning of tape.

Tape Writing:

PW [illegible]

0	PW	Prepare to write and write "file mark" at the beginning of a tape.
1	W or W_{10} or W_{12}	Write a block.

2	WFMR	Write "file mark" and prepare to read.
3	BW	Move backward one block or one file mark and prepare to write.
4	GAP	Erase about 6.5 inches of tape. (forward direction).
5	WFM	Write "file Mark".

Notes concerning tape operation:

Tape reading and writing can only be done when the unit is in the proper state. The unit must be in a "reading" state before any operation that does not write or erase information on tape may be done. The unit must be in a "writing" state before any operation of writing or erasing is done. A "Prepare to read (PR)" instruction must be done at the beginning of a tape that is going to be read. This positions the tape between the initial file mark and the first block of information. A "Prepare to write (PW)" instruction must be done at the beginning of a tape that is going to be written and this also writes a file mark on the tape. To switch from reading to writing at any other position of the tape, the "move backward one block or one file mark and prepare to write (BW)" order must be used.

To switch from writing to reading, the order "write file mark and prepare to read (WFMR)" must be used.

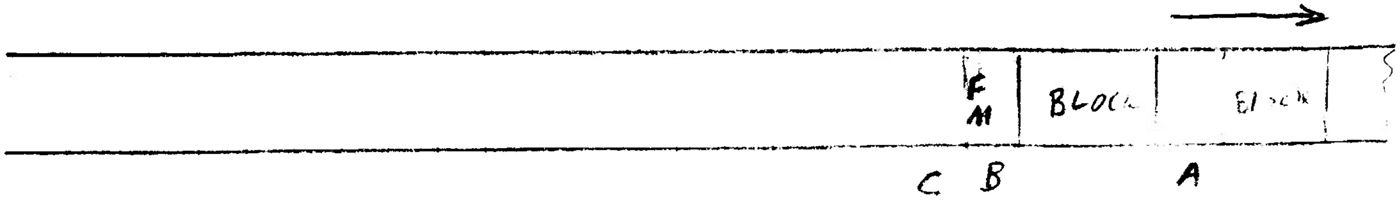
File marks are ignored by the tape "read" and "move B blocks" instructions.

Drum reading:

8	Clear B memory positions starting at A. (Does not wait for drum access.)

Drum writing:

8	Clear specified drum positions on the drum.



Changing [illegible] I will [illegible].

ASSUME:

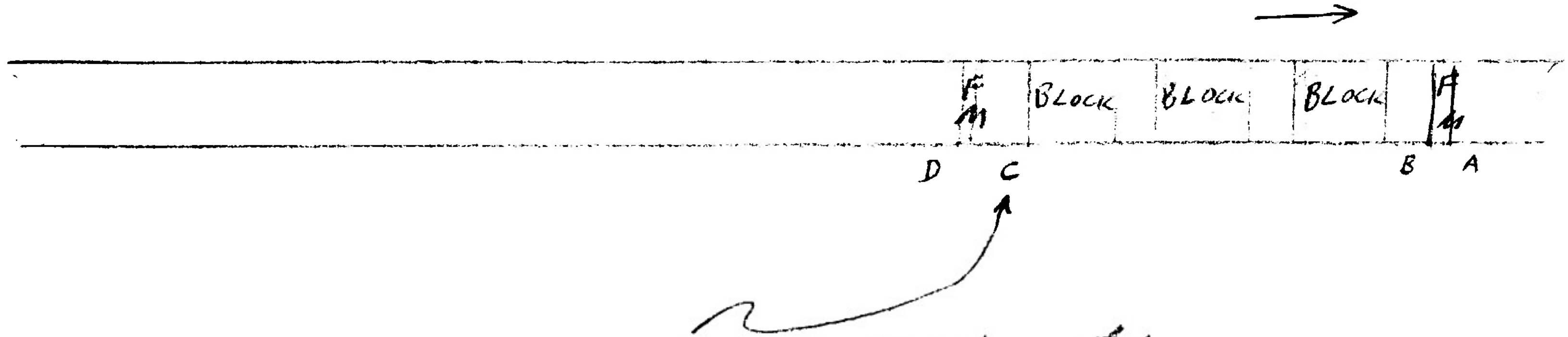
1. Not at beginning of tape.
2. READING [illegible] positioned at A or B.

MFMF [illegible] (from A or B) to C.

BW Tape moves [illegible] C to B. [illegible]

WFM [illegible].

(Con[illegible], In that case, when a block is written, the [illegible].)



Assume position C [illegible] (Writing state)

WFM[illegible] [illegible] D[illegible]

MFMB N[illegible] is C[illegible]

MFMB N[illegible] A

R12 [illegible]

c_4 and c_5 bits: Drum address within the channel at which to begin transferring words. Each channel will have at least 2,048 words and transfer can only be started at every 16th word on the drum.

Magnetic Tape:

$c_4 = 0$ Machine will halt at parity error.

$c_4 = 1$ P Machine will not halt at parity error.

b β (non-indexable) Bits for "Jump if condition" and "Jump if not condition" Instructions

If a bit in b β is a 1, that condition is tested and if "on" is turned "off". Bit P_{33} has a special meaning; if it is 1 the result of the last arithmetic instruction is included in the test. It may not be included if $P_{33} = 0$. (See Concurrent Operation.) If $P_{33} = 1$ and all other test bits are zero, then only the previous arithmetic order is tested for any possible error condition. This test however does not turn the error "off". The check of input-output conditions is always made at the time the test instruction is given which is not necessarily after the input-output instruction is finished. (See Concurrent Operation.)

Sexadecimal	bit	Condition to test when bit = 1.
2	P_{38}	Magnetic tape trunk B busy?
1	P_{37}	Magnetic tape trunk A busy?
8	P_{36}	Card Punch busy?
4	P_{35}	Card Reader busy?
2	P_{34}	Drum busy?

1	P_{33}	Special: Include result of previous arithmetic order in test or test only the previous arithmetic order if no other test bits are specified.
8	P_{32}	Has any tape unit using trunk B reached end of tape?
4	P_{31}	Has any tape unit using trunk A reached end of tape?
2	P_{30}	Has there been a tape parity error on trunk B?
1	P_{29}	Has there been a tape parity error on trunk A?
4	P_{27}	Has square root of a negative number been taken?
2	P_{26}	Has there been floating point division by zero or by an unnormalized divisor when doing normalized arithmetic?
1	P_{25}	Has there been floating point exponent overflow?
8	P_{24}	Has fixed point shift caused overflow?
4	P_{23}	Has fixed point divide caused overflow?
2	P_{22}	Has fixed point multiply caused overflow?
1	P_{21}	Has fixed point add or substract caused overflow?

Concurrent Operation

Except for arithmetic or boolean compare instructions, the test overflow instructions with $P_{33} = 1$, or any arithmetic order that stores in any index register or stores in the location of the next instruction, the

There are [illegible]

[illegible]

machine always gets its next instruction from the memory while it is doing the previous instruction. If this next instruction is one of the control and indexing orders (Basic Order Type = 0 or 1), it is immediately done, unless it is an input-output order or a test overflow order with $P_{33} = 1$. If it is done, it proceeds to get another instruction and do it if possible. Thus most all of the control and indexing orders can be done concurrently with the arithmetic or logical orders. Only the arithmetic and logical orders require the use of the main arithmetic unit of the machine. However the programmer must not attempt to use the result of the last arithmetic or logical orders in the "Jump if + (-)" or "Jump if (not) alphabetic character" orders and P_{33} must = 1 in any test overflow orders that are supposed to check the last arithmetic result for overflow.

All types of input-output orders can be done concurrently with other instructions. Automatic interlocks are provided so as to prevent timing conflict. Reference to a main memory position within the range or either an input or output instruction will halt the computer until the input or output transfer has occured at the memory position. The computer is released as soon as the transfer of that particular word has been made and does not wait for the entire transfer to be completed. There is no interlock on the index mamory when it is used as index registers (when in a, b, or c of an instruction), only the effective addresses A, B, C are conflict checked. Hence, if index memory is included in the range of an input-output instruction, no index register in that range should be used as such until the transfer of information has occurred. The programmer can easily make the computer wait until such a transfer is complete by using the last address in the index range of the in-out order in the A, B, or C addresses of a dummy order. An input-output instruction is not started until the previous arithmetic instruction is finished, hence the last arithmetic result may be included in the range of any input-output order.

As many as five input-output orders can be operating concurrently with computing and with each other. There is a separate trunk for reading cards, punching cards, using drum, and two separate trunks for using magnetic tape and all five of these trunks can operate concurrently. Input-output orders are not interlocked with each other and thus two or more may use the same memory cells.

Lloyd W. Campbell

LLOYD W. CAMPBELL

Glenn Beck.

GLENN BECK

DISTRIBUTION LIST

No. of Copies	Organization
10	Commander Armed Services Technical Information Agency ATTN: TIPCR Arlington Hall Station Arlington 12, Virginia
1	Director Advanced Research Projects Agency Department of Defense Washington 25, D.C.
1	Director, National Security Agency ATTN: R/D 36, Chief, Engineering Research Division Fort George G. Meade, Maryland
1	Chief of Ordnance ATTN: ORDTB - Bal Sec Department of the Army Washington 25, D.C.
1	Commanding General Frankford Arsenal Philadelphia, Pennsylvania
1	Commanding Officer Picatinny Arsenal ATTN: Feltman Research and Engineering Laboratories Dover, New Jersey
1	Commanding Officer Watertown Arsenal Watertown 72, Massachusetts
1	Commanding General Army Ballistic Missile Agency Redstone Arsenal, Alabama
1	Commanding General Army Rocket & Guided Missile Agency Redstone Arsenal, Alabama
1	Commanding Officer Diamond Ordnance Fuze Laboratories ATTN: Technical Information Office, Branch 041 Washington 25, D.C.
1	Commanding Officer Diamond Ordnance Fuze Laboratories Washington 25, D.C.
1	Commanding General U.S. Army Ordnance Missile Command Redstone Arsenal, Alabama
1	Commanding General Ordnance Special Weapons Ammunition Command Picatinny Arsenal Dover, New Jersey
1	Commanding General Ordnance Weapons Command Rock Island, Illinois
1	Commanding General White Sands Missile Range New Mexico
1	Director Land Locomotion Research Laboratory Ordnance Tank-Automotive Command 1501 Beard Street Detroit 9, Michigan
1	Director Major Item Supply Management Agency Letterkenny Ordnance Depot Chambersburg, Pennsylvania
1	Research Analysis Corporation ATTN: Computer Facility 6935 Arlington Road Bethesda, Maryland Washington 14, D.C.
1	Commanding General U.S. Army Chemical Corps Research & Development Command Washington 25, D.C.
1	Commanding Officer U.S. Army Chemical Warfare Labs ATTN: Dr. Carl M. Herget Army Chemical Center, Maryland

DISTRIBUTION LIST

No. of Copies	Organization
1	Commanding General Chemical Corps Materiel Command ATTN: David T. Shepard, Director Data Processing Center Army Chemical Center, Maryland
1	Commanding General U.S. Army Chemical Center Corps Engineering Command Army Chemical Center, Maryland
1	Director U.S. Army Nuclear Defense Laboratory Army Chemical Center, Maryland
1	Commanding Officer Operations Research Group Army Chemical Center, Maryland
1	Director U.S. Army Chemical Corps Quality Assurance Technical Agency Army Chemical Center, Maryland
1	Commanding Officer U.S. Army Chemical Research and Development Laboratories Army Chemical Center, Maryland
1	Chairman U.S. Army Chemical Corps Technical Committee Army Chemical Center, Maryland
1	Director U.S. Army Biological Laboratories Camp Detrick, Maryland
1	Commanding General U.S. Army Chemical Corps Proving Ground Dugway Proving Ground Dugway, Utah
1	Commanding Officer U.S. Army Corps of Engineers Army Reactors Group Fort Belvoir, Virginia
1	Director U.S. Army Corps of Engineers Geodessy Intelligence and Mapping Research & Development Agency Fort Belvoir, Virginia
1	Commanding Officer U.S. Army Corps of Engineers Polar Research & Development Center Fort Belvoir, Virginia
1	Commanding General Engineer Research & Development Labs U.S. Army Fort Belvoir, Virginia
1	Commanding Officer U.S. Army Corps of Engineers Army Reactors Group Germantown, Maryland
1	Commanding Officer U.S. Army Corps of Engineers Waterways Experiment Station P.O. Box 631 Vicksburg, Mississippi
1	U.S. Army Corps of Engineers Special Assistant for Nuclear Power Building T-7 Washington 25, D.C.
1	Director U.S. Army Corps of Engineers Cold Regions Research and Engineering Laboratory 1215 Washington Avenue Wilmette, Illinois

DISTRIBUTION LIST

No. of Copies	Organization
1	Director U.S. Army Medical Research and Nutrition Laboratory Denver, Colorado
1	Commanding Officer U.S. Army Medical Research and Development Command Washington 25, D.C.
1	Commanding Officer U.S. Army Medical Unit Camp Detrick, Maryland
1	Director U.S. Army Medical Research Laboratory Fort Knox, Kentucky
1	Director U.S. Army Quartermaster Research and Engineering Field Evaluation Agency ATTN: TD 10-5411-02 Fort Lee, Virginia
1	Commanding Officer U.S. Army Signal Engineering Agency Arlington Hall Station Arlington, Virginia
1	Commanding General U.S. Army Signal Missile Support Agency White Sands Missile Range New Mexico
1	Commanding Officer U.S. Army Signal Intelligence Agency Arlington Hall Station Arlington, Virginia
1	Commanding General U.S. Army Electronic Proving Ground Fort Huachuca, Arizona
1	Commanding Officer U.S. Army Signal Electronic Research Unit P.O. Box 205 Mountain View, California
1	Commanding Officer U.S. Army Signal Avionics Field Office P.O. Box 209 St. Louis 66, Missouri
1	Chief Signal Officer ATTN: Mr. G.H. McClurg - SIGRD - 6 Department of the Army Washington 25, D.C.
1	Commanding General U.S. Army Signal Research and Development Laboratory ATTN: Mr. David C. Law Data Equipment Branch Fort Monmouth, New Jersey
1	Commandant U.S. Army Signal Corps School ATTN: Officer Department Fort Monmouth, New Jersey
1	Commanding Officer U.S. Transportation Materiel Command 12th and Spruce Streets St. Louis, Missouri
1	Commanding General U.S. Army Transportation Research Command Fort Eustis, Virginia
1	Commanding General U.S. Continental Army Command Fort Monroe, Virginia

DISTRIBUTION LIST

No. of Copies	Organization
1	Commandant U.S. Army Artillery & Guided Missile School Fort Sill, Oklahoma
1	Commandant U.S. Army Ordnance Guided Missile School ATTN: Redstone Section, BM Branch, FAM Division Redstone Arsenal, Alabama
1	Commanding Officer Army Research Officer (Durham) Box CM, Duke Station Durham, North Carolina
1	Army Research Office Arlington Hall Station Arlington, Virginia
1	Commandant Command & General Staff College ATTN: Computing Facility Fort Leavenworth, Kansas
1	Commanding Officer U.S. Army Combat Surveillance Agency 1124 Highland Street Arlington, Virginia
1	Commanding Officer U.S. Army Communications Agency The Pentagon Washington 25, D.C.
1	Commanding General Fort George G. Meade, Maryland ATTN: Computing Facility
1	Professor of Ordnance U.S. Military Academy West Point, New York
1	Chief of Naval Operations Department of the Navy Washington 25, D.C.
3	Chief, Bureau of Naval Weapons ATTN: DIS-33 Department of the Navy Washington 25, D.C.
1	Commanding Officer and Director David W. Taylor Model Basin Washington 7, D.C.
1	Chief, Bureau of Yards & Docks ATTN: Data Processing and Analysis Branch Department of the Navy Washington 25, D.C.
1	Chief, Bureau of Ships ATTN: Computing Facility Department of the Navy Washington 25, D.C.
1	Superintendent U.S. Naval Academy ATTN: Weapons Department Annapolis, Maryland
1	Commanding Officer U.S. Naval Air Development Center Johnsville, Pennsylvania
1	Commanding Officer Naval Air Test Center ATTN: Armament Test U.S. Naval Air Station Patuxent River, Maryland
2	Commander U.S. Naval Missile Center ATTN: Simulation Branch Systems Department Range Operations Department, Code 3280 Point Mugu, California
1	Commander Naval Ordnance Laboratory White Oak, Silver Spring 19 Maryland

DISTRIBUTION LIST

No. of Copies	Organization
1	Commanding Officer Naval Ordnance Laboratory ATTN: G.E. Stewart Corona, California
1	Commander U.S. Naval Ordnance Test Station China Lake, California
1	Superintendent U.S. Naval Postgraduate School Monterey, California
1	Director U.S. Naval Research Laboratory ATTN: Mr. Nassetta, Code 492 Washington 25, D.C.
1	Commanding Officer & Director U.S. Naval Radiological Defense Laboratory San Francisco 24, California
1	Commander U.S. Naval Weapons Laboratory ATTN: Computation & Analysis Branch Dahlgren, Virginia
1	Director U.S. Naval Supersonic Laboratory Massachusetts Institute of Technology ATTN: Computer Facility 560 Memorial Drive Cambridge, Massachusetts
1	Commander Naval Engineering Experiment Station ATTN: Applied Math Office, Code 502 Annapolis, Maryland
1	Commander Arnold Engineering Development Center Tullahoma, Tennessee
1	Commander Air Force Flight Test Center ATTN: Data Processing & Computing Branch, FTFSD Edwards Air Force Base, California
1	Commander Air Force Cambridge Research Lab L.G. Hanscom Field Bedford, Massachusetts
1	Commander Missile Development Center Holloman Air Force Base, New Mexico
1	Commander Tactical Air Command ATTN: Director of Statistical Services Langley Air Force Base, Virginia
1	Director Air University Maxwell Air Force Base, Alabama
1	Commander Aeronautical Systems Division ATTN: Digital Computation Branch Wright-Patterson Air Force Base, Ohio
1	Commander U.S. Air Force Institute of Technology Wright-Patterson Air Force Base, Ohio
1	Superintendent United States Air Force Academy Colorado
1	Air Controller U.S. Air Force Washington 25, D.C.
2	Commanding Officer Air Force Intelligence Center Arlington Hall, Virginia
3	Director of Statistical Services ATTN: Mr. Joseph F. Cunningham Department of the Air Force Washington 25, D.C.
1	Brookhaven National Laboratory ATTN: Computer Facility Upton, New York

DISTRIBUTION LIST

No. of Copies	Organization
1	Director Air Weather Service Climatic Center 225 D Street, S.E. Washington 25, D.C.
1	Oak Ridge National Laboratory ATTN: Mr. E.C. Long P.O. Box X Oak Ridge, Tennessee
1	Central Intelligence Agency OCR/Library/ILS ATTN: Norman E. Hill, Code 163 Washington 25, D.C.
1	Federal Aviation Agency National Aviation Facilities Experimental Station ATTN: Simulation & Computation Br. Atlantic City, New Jersey
1	Federal Aviation Agency ATTN: Data Processing Branch - Aircraft Management Division Bureau of Flight Standards P.O. Box 1082 Oklahoma City, Oklahoma
1	Director National Aeronautics and Space Administration ATTN: Mr. R.E. Liettell 1520 H Street, N.W. Washington 25, D.C.
1	Director National Aeronautics and Space Administration Lewis Research Center ATTN: Computer Facility Cleveland Airport Cleveland, Ohio
1	Director National Aeronautics and Space Administration Flight Research Center ATTN: Computer Facility Box 273 Edwards, California

No. of Copies	Organization
2	Director National Aeronautics and Space Administration Goddard Space Flight Center ATTN: Tracking & Data Systems I. Mortimer Datz - Computer Operations Branch Data Systems Division Anacostia Naval Station Washington 25, D.C.
1	Director National Bureau of Standards ATTN: Dr. S.N. Alexander 232 Dynamometer Building Washington 25, D.C.
1	Director National Bureau of Standards ATTN: Mr. Paul Meissner Components & Techniques Section - Data Processing Systems Division 232 Dynamometer Building Washington 25, D.C.
2	Director National Bureau of Standards National Applied Mathematics Laboratory ATTN: Miss. Mary Stevens Dr. Franz L. Alt Computation Laboratory Washington 25, D.C.
1	U.S. Department of Commerce Bureau of Census ATTN: Computer Facility Federal Office Building No. 3 Suitland, Maryland
1	Engineering Research Associates Division of Remington Rand, Inc. 1902 W. Minnehaha Avenue St. Paul, Minnesota

DISTRIBUTION LIST

No. of Copies	Organization
2	Ampex Computer Products Co. 9937 Jefferson Boulevard Culver City, California
1	International Business Machine, Corporation Engineering Laboratory ATTN: John Ashley - Customer Executive Education Department San Jose, California
1	Raytheon Manufacturing Company P.O. Box 398 Bedford, Massachusetts
1	Remington Rand Univac Division of Sperry Rand Corporation ATTN: Mr. Sam Howry - Systems Analysis 1900 W. Allegheny Avenue St. Paul, Minnesota
2	Technitrol Engineering Corporation 1952 E. Alleghany Avenue Philadelphia 34, Pennsylvania
1	California Institue of Technology ATTN: R.B. Gilmore - Comptroller Pasadena, California
1	California Institue of Technology Jet Propulsion Laboratory ATTN: Computer Facility 4800 Oak Grove Drive Pasadena 3, California
1	Columbia University Electronics Research Laboratories ATTN: G.S. Bodeen 632 West 125 Street New York 27, New York
1	Columbia University Lewis Cyclation Laboratory ATTN: Computer Facility Box 137 Irvington on Hudson, New York
1	Cornell University ATTN: John W. Hastie - Coordinator of Research Ithaca, New York
1	Dartmouth College ATTN: Computation Center Hanover, New Hampshire
1	The George Washington University ATTN: Logistics Research Project 707 22nd Street, N.W. Washington 7, D.C.
1	Georgia Institute of Technology Engineering Experiment Station ATTN: Rich Electronic Computer Center Atlanta 13, Georgia
1	Harvard University Computation Laboratory Cambridge 38, Massachusetts
1	Indiana University ATTN: Research Computing Center Bloomington, Indiana
1	Iowa State University of Science and Technology Engineering Experiment Station ATTN: Robert M. Stewart, Jr. Cyclone Computer Lab Ames, Iowa
1	The Johns Hopkins University Applied Physics Laboratory ATTN: Computer Facility 8621 Georgia Avenue Silver Spring, Maryland
1	The Johns Hopkins University ATTN: Computation Center 34th and Charles Streets Baltimore 18, Maryland
1	Lehigh University ATTN: Computer Facility Bethlehem, Pennsylvania

DISTRIBUTION LIST

No. of Copies — Organization

1 Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Computer Facility
Lexington 73, Massachusetts

1 Massachusetts Institute of Technology
ATTN: Digital Computer Laboratory
Cambridge 39, Massachusetts

1 Michigan State University
College of Engineering
ATTN: M.G. Keeney - Computer Lab
East Lansing, Michigan

1 Marquette University
ATTN: Computing Center
1515 West Wisconsin Avenue
Milwaukee, Wisconsin

1 Missouri School of Mines and Metallurgy
ATTN: Computer Facility
Rolla, Missouri

1 Moore School of Electrical Engineering
University of Pennsylvania
Philadelphia 4, Pennsylvania

1 New York University
College of Engineering
ATTN: Computation & Statistical Lab
University Heights
New York 53, New York

1 Oklahoma State University
The Computing Center
ATTN: D.R. Shreve -
Department of Mathematics
Stillwater, Oklahoma

1 Oregon State College
Department of Mathematics
ATTN: W.E. Milne
Corvallis, Oregon

No. of Copies — Organization

1 Polytechnic Institute of Brooklyn
ATTN: H.P. Wile -
Administrator of Research
333 Jay Street
Brooklyn 1, New York

1 Princeton University
Mathematics Department
Princeton, New Jersey

1 Stanford University
ATTN: Computation Center
Stanford, California

1 University of California
ATTN: D.H. Lehmer
942 Hilldale Avenue
Berkeley, California

1 University of Illinois
Department of Mathematics
ATTN: A.H. Taub
Urbana, Illinois

1 R.F. Jackson
University of Delaware
Newark, Delaware

1 Dr. L.H. Thomas
Watson Scientific Computing Laboratory
612 W. 116th Street
New York 27, New York

10 Commander
British Army Staff
British Defence Staff (W)
ATTN: Reports Officer
3100 Massachusetts Avenue, N.W.
Washington 8, D.C.

4 Defence Research Member
Canadian Joint Staff
2450 Massachusetts Avenue, N.W.
Washington 8, D.C.

AD Accession No. UNCLASSIFIED
Ballistic Research Laboratories, APG
THE INSTRUCTION CODE FOR THE BRL
ELECTRONIC SCIENTIFIC COMPUTER (BRLESC)
Lloyd W. Campbell and Glenn Beck

Mathematical Computers - Coding
Mathematical Computers - Operation

BRLM Report No. 1379 November 1961

DA Proj No. 503-06-002, OMSC No. 5010.11.812
UNCLASSIFIED Report

The complete instructions repertoire of the BRL's new Electronic Scientific Computer (BRLESC) is given. Some engineering features are included.

AD Accession No. UNCLASSIFIED
Ballistic Research Laboratories, APG
THE INSTRUCTION CODE FOR THE BRL
ELECTRONIC SCIENTIFIC COMPUTER (BRLESC)
Lloyd W. Campbell and Glenn Beck

Mathematical Computers - Coding
Mathematical Computers - Operation

BRLM Report No. 1379 November 1961

DA Proj 503-06-002, OMSC No. 5010.11812
UNCLASSIFIED Report

The complete instructions repertoire of the BRL's Electronic Scientific Computer (BRLESC) is given. Some engineering features are included.

AD Accession No. UNCLASSIFIED
Ballistic Research Laboratories, APG
THE INSTRUCTION CODE FOR THE BRL
ELECTRONIC SCIENTIFIC COMPUTER (BRLESC)
Lloyd W. Campbell and Glenn Beck

Mathematical Computers - Coding
Mathematical Computers - Operation

BRLM Report No. 1379 November 1961

DA Proj No. 503-06-002, OMSC No. 5010.11.812
UNCLASSIFIED Report

The complete instructions repertoire of the BRL's new Electronic Scientific Computer (BRLESC) is given. Some engineering features are included.

AD Accession No. UNCLASSIFIED
Ballistic Research Laboratories, APG
THE INSTRUCTION CODE FOR THE BRL
ELECTRONIC SCIENTIFIC COMPUTER (BRLESC)
Lloyd W. Campbell and Glenn Beck

Mathematical Computers - Coding
Mathematical Computers - Operation

BRLM Report No. 1379 November 1961

DA Proj No. 503-06-002, OMSC No. 5010.11.812
UNCLASSIFIED Report

The complete instructions repertoire of the BRL's Electronic Scientific Computer (BRLESC) is given. Some engineering features are included.